



1.0

2.5

2.2

1.1

2.0

1.8

1.25

1.4

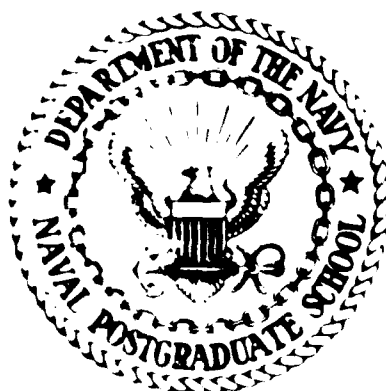
1.6

2

AD-A169 227

# NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC  
ELECTE  
JUL 02 1986  
S D

## THESIS

DETERMINATION OF SELECTIVE REENLISTMENT  
BONUS MULTIPLIERS IN THE  
UNITED STATES MARINE CORPS

by

Dean D. Dowd, Jr.

March 1986

For the Author:

R. Kevin Wolf

Approved for public release; distribution is unlimited.

DTIC FILE COPY

## REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b OFFICE SYMBOL (If applicable) Code 55		7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		8a NAME OF FUNDING/SPONSORING ORGANIZATION		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8b ADDRESS (City, State, and ZIP Code)		8b OFFICE SYMBOL (If applicable)		10 SOURCE OF FUNDING NUMBERS	
				PROGRAM ELEMENT NO	PROJECT NO
				TASK NO	WORK UNIT ACCESSION NO
11 TITLE (Include Security Classification) DETERMINATION OF SELECTIVE REENLISTMENT BONUS MULTIPLIERS IN THE UNITED STATES MARINE CORPS					
12 PERSONAL AUTHOR(S) DeWolfe, Dean D.					
13a TYPE OF REPORT Master's Thesis		13b TIME COVERED FROM TO		14 DATE OF REPORT (Year, Month, Day) 1986, March	
15 PAGE COUNT 50					
16 SUPPLEMENTARY NOTATION					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Math Programming, Integer Programming, Knapsack Problem, Selective Reenlistment Bonus, Lagrangian Relaxation, Generalized Assignment Problem		
19 ABSTRACT (Continue on reverse if necessary and identify by block number) Selective Reenlistment Bonuses (SRBs) are offered to improve retention in designated military occupational specialties (MOSs) for specified years-of-service intervals (zones). The amount of the bonus is set by assigning an "SRB Multiplier" for each MOS and zone combination (cell). Determination of multipliers is modeled as a nonlinear knapsack problem which is then linearized to a generalized assignment problem. The objective is to minimize the sum over all cells of a weighted squared deviation from the reenlistment target in each cell. Lagrangian relaxation provides lower bounds and feasible solutions. The best feasible solution is improved using a greedy heuristic to apportion unexpended funds. A FORTRAN 77 computer program implements the procedure. Data for FY86 yields a 0-1 integer program with 1795 binary variables and 980 constraints. A solution within .01 of optimality is obtained on an IBM 3033AP in 1.7 seconds and on an IBM PC in about four minutes.					
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL K. Kevin Wood			22b TELEPHONE (Include Area Code) (408) 646-2523		22c OFFICE SYMBOL Code 55Wd

Approved for public release; distribution is unlimited.

Determination of Selective Reenlistment Bonus Multipliers  
in the  
United States Marine Corps

by

Dean D. DeWolfe  
Captain, United States Marine Corps  
B.S., University of Virginia, 1980

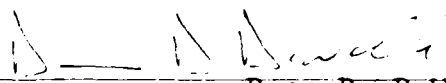
Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

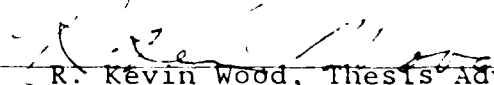
from the

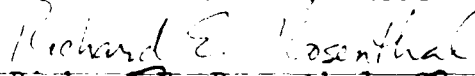
NAVAL POSTGRADUATE SCHOOL  
March 1986

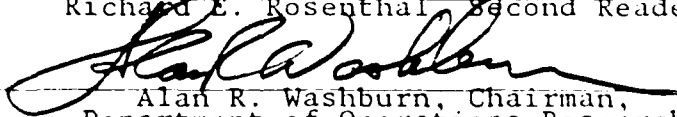
Author:

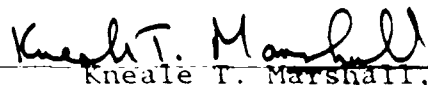
  
Dean D. DeWolfe

Approved by:

  
R. Kevin Wood, Thesis Advisor

  
Richard E. Rosenthal, Second Reader

  
Alan R. Washburn, Chairman,  
Department of Operations Research

  
Kneale T. Marshall,  
Dean of Information and Policy Sciences

# ABSTRACT

Selective Reenlistment Bonuses (SRBs) are offered to improve retention in designated military occupational specialties (MOSSs) for specified years-of-service intervals (zones). The amount of the bonus is set by assigning an "SRB Multiplier" for each MOS and zone combination (cell).

Determination of multipliers is modeled as a nonlinear knapsack problem which is then linearized to a generalized assignment problem. The objective is to minimize the sum over all cells of a weighted squared deviation from the reenlistment target in each cell. Lagrangian relaxation provides lower bounds and feasible solutions. The best feasible solution is improved using a greedy heuristic to apportion unexpended funds.

A FORTRAN 77 computer program implements the procedure. Data for FY86 yields a 0-1 integer program with 4895 binary variables and 980 constraints. A solution within .01% of optimality is obtained on an IBM 3033AP in 1.7 seconds and on an IBM PC in about four minutes.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability	
Dist	
A-1	

## TABLE OF CONTENTS

I.	INTRODUCTION . . . . .	6
	A. PROBLEM STATEMENT . . . . .	6
	B. PROBLEM SCOPE . . . . .	8
	C. MODEL AND SOLUTION APPROACH . . . . .	9
	D. THESIS OUTLINE . . . . .	10
II.	MODEL FORMULATION AND DESCRIPTION . . . . .	11
	A. MODEL DEVELOPMENT . . . . .	11
	B. CONVERSION TO A GENERALIZED ASSIGNMENT PROBLEM . . . . .	15
III.	SOLUTION METHODOLOGY . . . . .	18
	A. LAGRANGIAN RELAXATION TO ESTABLISH A LOWER BOUND . . . . .	18
	B. MARGINAL RATE OF RETURN HEURISTIC . . . . .	21
IV.	IMPLEMENTATION AND COMPUTATIONAL RESULTS . . . . .	23
	A. COMPUTER PROGRAM . . . . .	23
	B. TEST DATA . . . . .	24
	C. COMPUTATIONAL RESULTS . . . . .	24
V.	CONCLUSIONS AND RECOMMENDATIONS . . . . .	26
	APPENDIX A: LISTING OF SOURCE CODE . . . . .	28
	APPENDIX B: EXAMPLES OF INPUT FILES . . . . .	33
	APPENDIX C: SAMPLE FROM OUTPUT FILE . . . . .	39
	APPENDIX D: USER INSTRUCTIONS . . . . .	40
	A. DATA FILES . . . . .	40
	B. RUNNING THE PROGRAM . . . . .	44
	1. Checking Data Files for Conformity . . . . .	45
	2. Running SRBMULT . . . . .	45

C. EXOGENOUS WEIGHTING FACTORS. PRESET MULTIPLIERS AND THE OVER/UNDER FACTOR . . . .	46
LIST OF REFERENCES . . . . .	48
INITIAL DISTRIBUTION LIST . . . . .	49



## I. INTRODUCTION

The United States Marine Corps (USMC) uses Selective Reenlistment Bonuses (SRBs) to increase retention in designated military occupational specialties (MOSs) and years-of-service intervals (zones). Those MOSs and zones which will receive bonuses as well as the dollar amounts of the bonuses are determined by the assignment of "SRB Multipliers." In this study, the problem of assigning multipliers given a limited budget is modeled and solved as a mathematical program. In this chapter, the SRB program is described, terms are explained and the problem itself is defined. In addition, the solution approach is briefly discussed and the structure of this thesis is outlined.

### A. PROBLEM STATEMENT

The Marine Corps SRB program was developed as a reenlistment incentive to increase retention in designated MOSs. In effect, an SRB is a sum of money offered to individuals with specific skills and years in service to encourage them to reenlist and thereby selectively increase manning levels. The bonus is applied at reenlistment points that fall between 21 months and 14 years of active duty service (ADS). This period is divided into three reenlistment zones. As defined in the applicable Department of Defense (DoD) directive [Ref. 1], the period between 21 months and 6 years of ADS is designated as Zone A, the period between 6 and 10 years as Zone B and the period between 10 and 14 years as Zone C.

SRBs are not offered to everyone who reenlists, rather they are directed at selected MOS and zone combinations, called "cells" in this study. For each cell, there exists a reenlistment target which is defined as the desired number

of reenlistments in that cell. In many cells, enough Marines will reenlist without any added inducement so that the reenlistment targets will be automatically met or exceeded. In others, however, the absence of a bonus will result in insufficient reenlistments. Through the offer of bonuses to reenlist, it is possible to reach or approach the targets. To permit additional flexibility in awarding bonuses, the amount of an SRB can vary from a minimum of 0 to a maximum of \$16,000. From experience, it is known that larger bonuses result in greater retention rates although the rate at which retention increases with the size of the bonus differs greatly between cells. The total amount of bonus money that can be awarded per fiscal year is limited by the SRB program's annual budget. Since an individual receives 75% of the bonus as a lump sum payment with the remainder apportioned in equal annual payments over the term of the reenlistment, the total SRB budget must consist of two parts. The first is money owed from bonuses from previous years and its apportionment is non-discretionary. The relevant budget here is the remainder which can be allocated for current year bonuses.

Once the decision is made to consider the use of a non-zero SRB to improve manning in a particular cell, the actual monetary size of the bonus is based on increments of the eligible member's monthly base pay. The individual receives an amount equal to his monthly base pay multiplied by that cell's SRB multiplier and the number of years of additional obligated service, subject to a maximum of \$16,000. The DoD directive stipulates that the multipliers may not exceed 6 and may be fractional amounts. The Marine Corps' implementation of this policy is somewhat more restrictive. It chooses not to use fractional multipliers and does not permit a multiplier greater than 5 for reenlistments in Zone A, 4 in Zone B and 3 in Zone C. Assigning a multiplier of 0 is equivalent to no bonus.

Presently, SRB multipliers are assigned manually in an iterative procedure. Multipliers are selected for cells based on the desired number of reenlistments for those cells. Once a particular combination of SRB multipliers has been selected, a microcomputer based spreadsheet package is used to determine if the proposed combination is feasible, i.e., within the allowed budget. If it is not, multipliers are changed and the calculation is repeated.

At this time, there is no objective function with which alternative sets of proposed multipliers can be evaluated or compared. Instead, the officer responsible for assigning multipliers relies exclusively on his experience, judgement and various rules of thumb to judge the desirability of a particular set of multipliers. Even if a sensible objective function were defined, it is doubtful that an optimal solution could be determined using a manual procedure. Of the approximately 1000 cells, on average, 450 are assigned non-zero SRB multipliers during any given period. It is very likely that the dimensionality of the problem would quickly overwhelm any ability to manually arrive at an optimal allocation.

This study proposes an alternative procedure. It is automated due to the large amount of data. An objective function is defined and justified, thereby permitting the comparison of different sets of multipliers. Further, an optimization method built around this objective function and the budget constraint is described and tested. Using this, a nearly optimal set of SRB multipliers can be determined. This will result in a more effective allocation of the SRB program budget.

#### B. PROBLEM SCOPE

The goal of the SRB program is to reduce expected shortfalls in the number of reenlistments in particular cells by increasing the retention rate in those cells through the

offer of a reenlistment bonus. The model that is developed in this study tries to achieve that goal by determining a set of SRB multipliers which, by a selected definition, can be considered optimal. This is a single period model; no attempt is made to forecast sets of multipliers for subsequent years based on expected retention resulting from the current year's set of multipliers. It allocates that part of the budget not obligated for annual payments for bonuses from previous years and does not consider the effect of the 25% of the bonus that is apportioned in future years. Furthermore, the model, through its use of expected values, is essentially deterministic. Other assumptions regarding the model are explicitly addressed in the succeeding chapter.

#### C. MODEL AND SOLUTION APPROACH

In the mathematical formulation that is proposed to model the assignment of SRB multipliers, the objective function to be minimized is the sum of individual cell penalties. These penalties are nonlinear increasing functions in the deviation from the reenlistment target for the cell. Inclusion of the budget constraint, which is also nonlinear, results in a nonlinear knapsack problem. The nonlinearity, coupled with other complicating factors, prevents the problem from being efficiently solved using common knapsack algorithms. For descriptive and computational purposes, the problem is reformulated as a special generalized assignment problem.

In this generalized assignment formulation, cells represent the tasks that must be assigned. Unlike conventional generalized assignment problems, there is but a single agent who must accomplish all the tasks but who has alternative processes for each. These alternative processes correspond to the various multiplier levels permitted for that cell.

Because of the special structure of this problem, Lagrangian relaxation is particularly well suited as a solution technique. Lagrangian relaxation has been successfully used in scheduling (e.g., [Ref. 2]), location (e.g., [Ref. 3]) and set-covering problems (e.g., [Ref. 4]) as well as generalized assignment problems [Ref. 9]. Of interest is that previous applications to generalized assignment problems have shared the characteristic of multiple tasks and multiple agents; a review of the literature has not revealed an application with a formulation similar to the one developed in this study.

An important advantage to the Lagrangian relaxation approach is that, in the process of establishing a lower bound on the optimal solution, feasible solutions are also obtained. A heuristic method is developed which improves the best feasible solution uncovered in the bounding procedure. The heuristically improved set of multipliers is accepted as the final solution provided its objective value is sufficiently close to the lower bound.

#### D. THESIS OUTLINE

This thesis develops and presents a method for determining SRB multipliers in the USMC. In Chapter II, the problem is formulated first as a nonlinear knapsack problem. Then, to facilitate the description of the solution approach and for computational purposes, it is reformulated as a special generalized assignment problem. The solution methodology is presented in Chapter III. The approach uses the technique of Lagrangian relaxation combined with a heuristic procedure to provide a final solution. Details on the implementation of this process and computational results are presented in Chapter IV. Conclusions and recommendations are contained in Chapter V. Lastly, a listing of the source code, user instructions and samples from input and output files are included as appendices.

## II. MODEL FORMULATION AND DESCRIPTION

In this chapter, the process of determining multipliers for all cells is formulated first as a simple nonlinear integer program. Coefficients are defined and described and the rationale behind the selection of the objective function is explained. For explanatory and computational purposes, the problem is then converted to a special case of the generalized assignment problem.

### A. MODEL DEVELOPMENT

The following formulation is developed to model the assignment of SRB multipliers:

Indices:

$i = 1, \dots, m$                   cells

Data:

$B$                                   budget

Functions:

$c_i(x_i)$                           penalty for setting multiplier at  $x_i$   
in cell  $i$

$r_i(x_i)$                           cost of setting multiplier at  $x_i$   
in cell  $i$

$n_i$                                   maximum permissible multiplier for  
cell  $i$ .  $n_i = 5, 4, 3$  for Zone A, B, C  
respectively

Decision Variables:

$x_i$                                   multiplier for cell  $i$

Formulation:

$$\min \sum_i c_i(x_i) \quad (P1)$$

$$\text{s.t. } \sum_i r_i(x_i) \leq B$$

$$x_i = 0, i_1 \leq i \leq i_2 \quad \text{if } i_1 \leq i \leq i_2 \\ x_i \text{ integer}$$

In deriving the objective function it is assumed that each cell's effect on the overall objective function is independent of all other cells. The objective function, then, is just the sum of individual cell penalties where  $r_i(x_i)$  should be defined in the same manner as Marine  $i$  is expected to reenlist given a  $k$ B multiplier  $x_i$ .  $r_i(x_i)$  should rise as the expected deviation from the reenlistment target increases. An individual term in the objective constraint represents the expected net usage requirements associated with selecting a particular multiplier value for that cell. For this formulation the following functional form was selected:

$$r_i(x_i) = \frac{E_i A_i^{1/2} W_i x_i}{E_i A_i^{1/2} x_i^{1/2} + D_i + K_i (x_i - E_i)^{1/2}} \quad \text{eqn 2.1}$$

$$r_i(x_i) = K_i x_i (E_i + \min\{0, 2P_i S_i x_i + 1200\})^{1/2} \quad \text{eqn 2.2}$$

where

- $T_1$  normalized cost of training a Marine in cell  $i$
- $A_i$  number of Marines in cell  $i$
- $V_i$  exogenous cell weighting factor
- $D_i$  reenlistment target for cell  $i$
- $R_i(x_i)$  reenlistment response rate in cell  $i$  when multiplier  $x_i$  is offered
- $E_i$  number of Marines in cell  $i$  eligible for reenlistment
- $P_i$  average monthly base pay of a Marine in cell  $i$
- $S_i$  duration of reenlistment of a Marine in cell  $i$
- $Q$  relative weight of overages to shortages with respect to the reenlistment target
- $(a \ b)^+$  positive part of  $(a \ b)$

The function  $n_i(x_i)$  represents a weighted squared deviation from the target number of reenlistments that would be expected in cell  $i$  if multiplier  $x_i$  were offered. It is the product of two sets of terms. The first term  $T_i A_i^{-1} W_i$  represents a "weight" assigned to each cell. A weighting scheme is required as deviations in some cells are considered more critical than in others. The second set of terms

$$[P_i [D_i - R_i(x_i)E_i]^2 + Q[(R_i(x_i)E_i - D_i)^2]$$

is the squared deviation from the desired number of reenlistments in cell  $i$  but where  $Q$  may be used to weight the penalty incurred by overages (the second difference term) differently from the base penalty incurred by shortages (the first difference term). The  $Q$ , the "over/under" factor, has been set to 1 in this study but values somewhat less than 1 might be appropriate.

Squaring the deviation has the effect of penalizing large deviations much more severely than smaller ones. This is appealing because it evens out shortages and overages among cells. Most would agree that it is less detrimental to the force as a whole to "spread the shortages around" than to have a potentially debilitating shortfall in a few cells. Spreading overages around is desirable for logistical reasons.

The deviation terms are calculated by squaring the difference between the desired number of reenlistments  $D_i$  and the expected number of reenlistments  $R_i(x_i)E_i$ . The function  $R_i(x_i)$  is discrete and defined only at integer values of  $x_i$  on the interval  $[0, n_i]$ . Specifically, when the individuals in cell  $i$  are offered an SRB multiplier  $x_i$ , a fraction  $R_i(x_i)$  of the  $E_i$  individuals eligible to reenlist will choose to do so. The values of  $R_i(x_i)$  have been estimated from responses to previous bonus offers; more sophisticated methods for determining these functional values, for



example, by including the effect of past and present economic conditions, are currently being studied [Ref. 5].

As stated previously, otherwise identical deviations in different cells are not viewed equally. Generally it costs the Marine Corps considerably more to replace or train an individual in a highly technical MOS than in a less technical MOS. Further, the actual number of Marines in a cell influences the acceptability of shortages. For example, a shortfall of 5 in a cell containing 1000 could be more easily tolerated than that same shortfall in a cell containing only 10.

Therefore, each cell  $i$  is assigned the weighting coefficient  $T_i A_i^{-1} W_i$ .  $T_i$  is the "training index" for cell  $i$  which is just the cost of training a Marine in cell  $i$  normalized by the maximum training cost over all cells.  $A_i^{-1}$  is the reciprocal of the actual number of Marines in cell  $i$ .  $W_i$  is an exogenous weighting factor which permits the user of the model, if desired, to exercise his judgement in weighting cells or blocks of cells; presently, all  $W_i$  are set to 1.

The form of the resource function  $r_i(x_i)$  is straightforward. This function represents the expected number of dollars that must be allocated in order to offer cell  $i$  an SRB multiplier of  $x_i$  and is the product of the expected number of reenlistments in cell  $i$  that would result from offering a bonus  $x_i$ ,  $R_i(x_i)E_i$ , and the size of the bonus  $\min\{(.75)P_i S_i x_i, \$12,000\}$  where  $\$12,000$  is 75% (the percentage of the bonus that the individual receives as a lump sum payment) of the maximum permissible bonus amount of  $\$16,000$ .

Problem P1 is recognizable as a nonlinear knapsack problem in which cells represent commodities and the budget represents the weight or cube constraint. However, the nonlinearity in the objective function and budget constraint make solution by standard branch and bound techniques

impossible [Ref. 6], and make solution by dynamic programming inefficient since standard reduction techniques [Ref. 7] are inapplicable. A straightforward dynamic programming solution is possible using the cells as stages and dollars as the sole state variable. However, significant computational difficulties would arise due to the large number of stages (on the order of  $10^3$ ) combined with a state variable that for FY86 can achieve any integer value on the interval  $[0, 7 \times 10^4]$ . The solution methodology that is used could be applied directly to the nonlinear knapsack formulation. However, the explanation of the implementation of this methodology will be more transparent if the problem is first converted to a linear integer problem, in particular, a generalized assignment problem. The following section describes this new formulation.

#### B. CONVERSION TO A GENERALIZED ASSIGNMENT PROBLEM

A typical formulation for the generalized assignment problem is: [Ref. 8]

Indices:

$k = 1, \dots, n$       agent  
 $i = 1, \dots, m$       task

Data:

$c_{ik}$                       cost if task  $i$  is assigned to agent  $k$   
 $r_{ik}$                       amount of resource expended if task  $i$  is assigned to agent  $k$   
 $a_k, b_k$                   minimum and maximum amounts of the resource that may be expended by agent  $k$

Decision Variables:

$x_{ik}$                       1 if task  $i$  is assigned to agent  $k$   
                              0 otherwise

Formulation:

$$\min \sum_i \sum_k c_{ik} x_{ik} \quad (P2)$$

$$\text{s.t. } a_k \leq \sum_i r_{ik} x_{ik} \leq b_k \quad k = 1, \dots, n$$

$$\sum_k x_{ik} = 1 \quad i = 1, \dots, m$$

$$x_{ik} \in \{0,1\}$$

In contrast to the classical assignment problem, the first  $n$  constraints imply that more than one task  $i$  can be assigned for completion to agent  $k$  provided that the resource constraints on agent  $k$  are not violated. (Only one type of resource is allowed.) The second set of  $m$  constraints requires that each task be assigned exactly once.

In the SRB multiplier selection problem the model is simplified to the following:

Indices:

$i = 1, \dots, m$  cells

$j = 0, \dots, n_i$  SRB multiplier values

Data:

$c_{ij}$   $c_i(j)$

$r_{ij}$   $r_i(j)$

$B$  budget

$n_i$  maximum permissible multiplier for cell  $i$   
 $n_i = 5, 4, 3$  for Zone A, B, C, respectively

Decision Variables:

$x_{ij}$  1 if multiplier  $j$  is selected for cell  $i$   
 0 otherwise

Formulation:

$$\min \sum_i \sum_j c_{ij} x_{ij} \quad (\text{P3a})$$

$$\text{s.t. } \sum_i \sum_j r_{ij} x_{ij} \leq B$$

$$\sum_j x_{ij} = 1 \quad i = 1, \dots, m$$

$$x_{ij} \in \{0,1\}$$

Here, a single agent is required to perform all  $m$  tasks but has at his disposal  $n_i$  alternative processes for each. The above formulation is completely equivalent to P1 although a continuous, i.e., linear programming (LP), relaxation of this formulation might be very weak since neither  $r_i(x_i)$  nor  $c_i(x_i)$  will typically be convex functions. No problems arise, however, since no LP relaxation is employed in the solution procedure.

This formulation of the model is quite general. The coefficients  $c_{ij}$  and  $r_{ij}$  are directly calculated by evaluating the functions  $c_i(x_i)$  and  $r_i(x_i)$  of the preceding section at all feasible values of  $x_i$ . Provided that the assumption of independence between cells is retained, alternative functions are permissible. It will be seen in the succeeding chapter that the solution methodology is equally general and introduction of alternative functions is trivial.

### III. SOLUTION METHODOLOGY

With the problem and formulation now defined, this chapter details the approach taken to find a satisfactory set of SRB multipliers. Commonly, bounds for integer programs are established using an LP relaxation; the special structure of this problem and the likely weakness of the LP relaxation, however, suggests an alternative approach using Lagrangian relaxation. The relaxed formulation is extraordinarily easy to solve as an integer program and the resulting bound is at least as good as that produced using an LP relaxation. Furthermore, in the course of determining the bound, feasible sets of multipliers are encountered, the best of which is heuristically improved to yield a final solution.

#### A. LAGRANGIAN RELAXATION TO ESTABLISH A LOWER BOUND

The technique of Lagrangian relaxation is well suited for those integer programming problems that would be relatively easy to solve were it not for one or more complicating constraints. Moreover, it has been successfully and efficiently applied previously to generalized assignment problems [Refs. 8,9].

As described in Fisher's excellent primer on Lagrangian relaxation [Ref. 10], the approach is to move complicating constraints into the objective function using the product of the Lagrangian multiplier and the constraint violation as a penalty term. To demonstrate, consider the linear integer programming problem:

$$\begin{array}{ll} \min & cx \\ \text{s.t.} & Ax \leq b \\ & Dx \leq e \\ & x \geq 0 \text{ and integer} \end{array} \quad (P4)$$

where the constraints  $Ax \leq b$  are the complicating ones. This is relaxed to:

$$\begin{aligned} \min_x \quad & cx + \lambda(Ax - b) \\ \text{s.t.} \quad & Dx \leq e \\ & x \geq 0 \text{ and integer} \end{aligned} \quad (P5a)$$

where it is assumed that this is easy to solve for a fixed row vector  $\lambda \geq 0$ . For such a  $\lambda$ , an optimal solution to problem P5 provides a lower bound on the objective value for P4. The best possible lower bound from a relaxation of the form of P5 is found by solving:

$$\max_{\lambda \geq 0} \left\{ \begin{array}{l} \min_x \quad cx + \lambda(Ax - b) \\ \text{s.t.} \quad Dx \leq e \\ \quad \quad x \geq 0 \text{ and integer} \end{array} \right\} \quad (P5b)$$

Furthermore, this bound is at least as good as that provided by an LP relaxation [Ref. 11].

Examination of the generalized assignment formulation developed in the preceding chapter indicates that the only truly complicating constraint is the budget constraint; integer optimization with this constraint relaxed consists of  $m$  separable, "multiple choice" problems which are trivial. Thus, the relaxed formulation with scalar  $\lambda$  is:

$$\max_{\lambda \geq 0} \left\{ \begin{array}{l} \min_x \quad \sum_{i,j} c_{ij} x_{ij} + \lambda (\sum_{i,j} r_{ij} x_{ij} - B) \\ \text{s.t.} \quad \sum_j x_{ij} = 1 \quad i = 1, \dots, m \\ \quad \quad x_{ij} \in \{0, 1\} \end{array} \right\} \quad (P3b)$$

where the inner portion of the objective function may be equivalently written:

$$\min_x \quad \sum_{i,j} x_{ij} (c_{ij} + \lambda r_{ij}) - \lambda B \quad (\text{eqn 3.1})$$

For a fixed  $\lambda$ , the inner minimization is performed by selecting a minimum  $(c_{ij} + \lambda r_{ij})$  for each cell. The outer maximization problem is a convex optimization problem which is easily solved, since  $\lambda$  is a scalar variable, by first bracketing  $\lambda$  and then performing a bisection search.

The bracket about the optimal  $\lambda$  is readily established. The lower bound  $L_l$  is zero. For  $\lambda$  sufficiently large, the solution to the inner minimization is to expend as few dollars as possible, i.e.,  $x_{i0}=1$  for all  $i$ .  $L_u$ , the upper bound on interesting values of  $\lambda$ , is the smallest value of  $\lambda$  for which this solution is optimal. Therefore,  $L_u$  is the smallest value of  $\lambda$  satisfying:

$$c_{i0} + \lambda r_{i0} \leq c_{ij} + \lambda r_{ij} \quad \forall i, j \geq 1 \quad (\text{eqn 3.2})$$

Recalling that  $r_{ij}$  represents the cost of offering cell  $i$  an SRB multiplier  $j$  and that  $j=0$  corresponds to no bonus,  $r_{i0}=0$  for all  $i$ . Thus, equation 3.2 becomes:

$$c_{i0} \leq c_{ij} + \lambda r_{ij} \quad \forall i, j \geq 1 \quad (\text{eqn 3.3})$$

and, as a result,

$$L_u = \max\{(c_{i0} - c_{ij})/r_{ij}\} \quad \forall i, j \geq 1 \quad (\text{eqn 3.4})$$

For  $\lambda = L_u$ , a tie exists between the solution  $x_{i0}=1$  for all  $i$  and at least one solution where  $x_{ij}=1$  for some  $i$  and some  $j \geq 1$ . For unusual data, the latter solution could be infeasible. To ensure that at least one feasible solution is obtained, in practice  $L_u$  is replaced by  $(1+\epsilon)L_u$  where  $\epsilon > 0$ .

The solution to P3b provides a lower bound on the optimal solution to the SRB multiplier problem. Upper bounds are provided by feasible sets of SRB multipliers encountered while solving P3b. Such solutions arise since, for  $\lambda$

sufficiently large, the entire budget will not be consumed. The best of the feasible sets of SRB multipliers, that is, the one with the smallest objective function value and, correspondingly, the lowest upper bound value, is improved using the heuristic procedure described in the following section.

#### B. MARGINAL RATE OF RETURN HEURISTIC

It seems reasonable to expect that the best feasible solution obtained in the process of optimizing P3b would require the expenditure of nearly the entire SRB budget. There might, however, be some budget dollars remaining, a residual that could be apportioned to further reduce the objective value of the best feasible solution which is the present upper bound. Accordingly, a heuristic method for allocating this residual was developed using the concept of marginal rate of return.

Given a best feasible set of SRB multipliers from the Lagrangian procedure, the incremental cost for each cell that would be realized by increasing that cell's multiplier by 1 is calculated for all those cells with multipliers not already at their maximum values. If this incremental amount is less than the unallocated portion of the budget it is possible to increase that cell's multiplier by 1 while remaining feasible. Let such a cell have  $x_{ij}=1$ . Then, the marginal rate of return for the cell is defined to be:

$$ROR_i = (c_{ij} - c_{i,j+1}) / (r_{i,j+1} - r_{ij}) \quad (\text{eqn 3.5})$$

This quotient expresses the improvement realized in the objective function per dollar spent when cell i's SRB multiplier is increased. In the heuristic procedure, the cell with the maximum positive  $ROR_i$  has its multiplier incremented. The residual portion of the budget is reduced and the process is repeated until it is no longer possible to



increase the multiplier of any cell. The resulting set of SRB multipliers is accepted as the final solution and upper bound for the SRB multiplier problem provided that the ratio between the lower and upper bound is close to 1.

Other heuristic improvement procedures are possible but were not implemented in this study. For example, by simultaneously incrementing one cell's SRB multiplier by 1 and decrementing another's (an  $O(m^2)$  operation) it might be possible to improve on the upper bound while remaining feasible. In fact, the problem could be solved to optimality using a branch and bound algorithm [Ref. 9] though the large number of decision variables might make this a slow process. Pursuing strict optimality, with the resultant increase in algorithmic complexity and solution times was not deemed worthwhile in light of the consistently good solutions that have been obtained.

#### IV. IMPLEMENTATION AND COMPUTATIONAL RESULTS

A FORTRAN 77 computer program was written to implement the solution methodology described in the preceding chapter. Using data obtained from the Manpower Department at Headquarters, USMC (HQMC), the model was run on both an IBM 3033AP and an IBM PC. To judge the robustness of the model, additional testing was performed using randomly perturbed data.

##### A. COMPUTER PROGRAM

The program that performs the Lagrangian bounding procedure and which yields the heuristically improved final solution was written in ANSI standard FORTRAN 77. No functions unique to the system were used. When the program was run on the IBM 3033AP it was compiled by the IBM VS FORTRAN compiler at OPT(2); on the IBM PC it was compiled using Ryan-McFarland's RM/FORTRAN compiler with code optimization. The program requires six input files: Parameter file, Reenlistment Plan file, Reenlistment Rates file, Training Cost file, Cell Size file, and a Cell Weight file. File specifications are detailed in Appendix D. As output, the program produces a file containing an MOS/Zone listing with the proposed SRB multipliers. For comparison purposes, the objective value as well as the percent deviation between the final solution's objective value and the lower bound on the optimal objective are output.

Under some circumstances the user may wish to preassign the SRB multipliers in specific cells. For example, the user may wish to declare the multipliers for all three zones in MOS 2112 to be some specified value. This may be accomplished by "tagging" those cells in the Cell Weight file and entering the desired SRB multiplier in the appropriate

column in the data file. Additional details are provided in Appendix D. Comparison of the objective values before and after the introduction of these preassigned multipliers enables the user to gauge the effect of *a priori* multiplier assignments.

#### B. TEST DATA

The FY86 data that was used to develop and test the model was obtained from the Enlisted Plans Section of the Manpower Department at HQMC. Model input consists of those values listed in equation 2.1 and 2.2 organized into five data files and one problem parameter file. Because the relevant data is extracted from a variety of sources, it was deemed desirable to place them in different files. The information contained within the Parameter file is primarily run specific. Samples from the various input files are contained in Appendix B.

#### C. COMPUTATIONAL RESULTS

The data described above leads to a problem with 979 cells. This in turn results in a problem, in the form of P3b, with 4895 binary variables and 980 constraints. The objective function value from the final solution is within .01% of the lower bound established through the Lagrangian procedure. In the process, 99.99% of the budget was expended. Appendix C contains a sample from the output file.

Running time on the IBM 3033AP was 1.7 seconds. 1.4 of the 1.7 seconds is used to read in the data and calculate the coefficients  $c_{ij}$  and  $r_{ij}$ . The Lagrangian procedure and the heuristic use most of the remaining .3 seconds. The time required to write the solution was negligible.

After preliminary development and testing on the IBM 3033AP, the program and data were copied onto a floppy diskette and, using the RM/FORTRAN software, installed on an IBM

PC configured with 512Kb RAM, an 8087 Math Coprocessor and two 360Kb floppy disk drives. No changes to the code were required in moving it to the PC. On the IBM PC, the program runs in approximately 4 minutes. Most of this 4 minutes is consumed reading in the data from the floppy diskette; slightly faster times might be expected using a system with a hard disk arrangement.

Additional testing was conducted to verify the robustness of the solution procedure and to confirm that the excellent results that were observed were not merely fortunate happenstance stemming from a "good" set of data. In this procedure, the data read by the model was perturbed in this manner:

$$d' = \lfloor d * UNIF(.5, 1.5) \rfloor$$

where

$d'$	perturbed datum
$d$	original datum
$UNIF(a, b)$	random variate generated from the continuous uniform distribution between $a$ and $b$
$\lfloor x \rfloor$	largest integer not exceeding $x$

This randomization process was performed on all the data except that response rates were capped at 1 and not discretized.

In all, 50 additional model runs were conducted in this manner, each with a different data set. Solution times on the IBM 3033AP varied slightly, ranging between 1.5 and 1.9 seconds. In each case, the final solution was within .02% of the lower bound on the optimal solution. Typically, the heuristic improves the best feasible solution with respect to the lower bound by less than .04%. In a situation in which exceedingly tight bounds are not necessary, the heuristic could, in fact, be eliminated.

## V. CONCLUSIONS AND RECOMMENDATIONS

In this thesis, the process of determining Selective Reenlistment Bonus multipliers subject to a limited budget was modeled as a mathematical program. In the course of doing this, an objective function based on expected deviations from cell reenlistment targets was defined. This is an important advance; formerly there was no method by which alternative sets of multipliers could be compared. Using this objective function, a procedure was developed which determines a nearly optimal set of SRB multipliers. The speed with which the solution is found, a modest core storage requirement and a compact, specialized FORTRAN code proves the procedure amenable for PC use. This is an important advantage given the PC's wide-spread availability. The inclusion of an exogenous weighting factor and the capability to preset multipliers allows the user additional flexibility in using the model.

The model and solution method that are employed are very general and only require that the assumption of inter-cell independence be maintained. Other penalty and resource utilization functions are readily incorporated into the model. Improvements in the estimates for the response rates should further increase the accuracy with which the model's multipliers achieve the cell targets.

Provided the user possesses the requisite hardware (the IBM PC configuration explained in Chapter IV should be considered the minimum) and a FORTRAN 77 compiler, the program is completely operational. A copy of the source code is listed in Appendix A. Other pertinent information, including examples of input and output files and user instructions, is contained within Appendices B, C, and D.

The future portends increasingly restrictive budgets within the Department of Defense and additional pressure to maximize benefits given limited resources. With respect to maximizing the benefits realized from the SRB program, a nearly optimal assignment of SRB multipliers goes far towards achieving that goal.

APPENDIX A  
LISTING OF SOURCE CODE

```

IMPLICIT REAL* 8 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*8 C(1100,0:5),A(1100,0:5)
INTEGER*2 MOS(1100),IZONE(1100),IXBEST(1100),IX(1100)
COMMON CINF,BUDGET,TEST,NUMCEL
CINF=1.D22
CALL INITAL(C,A,HLAM,MOS,IZONE,IX)
CALL BOUND(ZL,IX,ZU,IXBEST,HLAM,C,A,COST,MOS)
CALL HEURIS(C,A,IXBEST,ZH,COST,MOS)
WRITE (16,50)
50  FORMAT (1X,'*****')
PCTDEV=100.D0*(1.D0-ZL/ZH)
WRITE (16,100)ZH
100  FORMAT (1X,'OBJECTIVE FUNCTION VALUE IS:',F11.3)
WRITE (16,125)PCTDEV
125  FORMAT (1X,'THIS SOLUTION IS WITHIN ',F11.8,
* '% OF THE OPTIMAL')
WRITE (16,50)
WRITE (16,300)
DO 150 I=1,NUMCEL
    WRITE (16,200)MOS(I),IZONE(I),IXBEST(I)
150  CONTINUE
200  FORMAT (1X,I5,3X,I2,9X,I2)
300  FORMAT (3X,'MOS',3X,'ZONE',5X,'MULTIPLIER')
400  STOP
END
SUBROUTINE INITAL(C,A,HLAM,MOS,IZONE,IX)
IMPLICIT REAL* 8 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*8 AOS(3),APAY(3),C(1100,0:5),A(1100,0:5),R(0:5)
INTEGER*2 MOS(1),IZONE(1),IX(1)
COMMON CINF,BUDGET,TEST,NUMCEL
READ (10,100)BUDGET
READ (10,200)NUMCEL
READ (10,100)TIMAX
READ (10,300)(AOS(I),I=1,3)
READ (10,300)(APAY(I),I=1,3)
READ (10,100)BNSMAX
READ (10,400)PCTGFY
READ (10,400)Q
HLAM=-CINF
SCALE=1000.D0
BUDGET=BUDGET/SCALE
BNSMAX=BNSMAX*PCTGFY
TEST=CINF/1.1D0
READ(11,500)(MOS(I),IZONE(I),IX(I),A(I,0),I=1,NUMCEL)
READ (14,600)(A(I,1),A(I,2),I=1,NUMCEL)
DO 50 I=1,NUMCEL
    MAXMUL=6-IZONE(I)
    READ (15,700)(C(I,J),J=0,MAXMUL)
50  CONTINUE
    READ (12,800)(A(I,3),I=1,NUMCEL)
    READ (13,900)(A(I,4),I=1,NUMCEL)
DO 150 I=1,NUMCEL
    MAXMUL=6-IZONE(I)
    WT=A(I,0)
    E=A(I,1)
    TE=A(I,2)
    ACT=A(I,3)

```

```

TI=A(I,4)
IF (TI.LT. 1.DO) TI=50.DO
DO 75 J=0,MAXMUL
  R(J)=C(I,J)
CONTINUE
AVGPAY=APAY(IZONE(I))
AVGAOS=AOS(IZONE(I))
IF (ACT.GT. 0.0DO) THEN
  CW=WT*(TI/TIMAX)/ACT
ELSE
  CW=WT*TI/TIMAX
END IF
IF (E.LT. 1.DO) E=1.DO
DO 125 J=0,5
  IF (J.LE. MAXMUL) THEN
    DIFTRM=TE-E*R(J)
    IF (DIFTRM.LE. 0.DO) THEN
      C(I,J)=CW*Q*(DIFTRM**2)
    ELSE
      C(I,J)=CW*(DIFTRM**2)
    END IF
    BONUS=PCTGFY*AVGPAY*AVGAOS*J
    IF (BONUS.GT. BNSMAX) BONUS=BNSMAX
    A(I,J)=R(J)*E*BONUS/SCALE
    IF (J.GT. 0) THEN
      HL=(C(I,0)-C(I,J))/A(I,J)
      IF (HL.GT. HLAM) HLAM=HL
    END IF
  ELSE
    C(I,J)=CINF
    A(I,J)=CINF
  END IF
CONTINUE
125 CONTINUE
150 CONTINUE
100 FORMAT (F12.0)
200 FORMAT (I4)
300 FORMAT (3F7.0)
400 FORMAT (F4.2)
500 FORMAT (I5,2X,I1,2X,I1,2X,F4.2)
600 FORMAT (8X,2F10.0)
700 FORMAT (8X,6F6.2)
800 FORMAT (8X,F6.0)
900 FORMAT (8X,F11.0)
RETURN
END
SUBROUTINE BOUND(ZL,IX,ZU,IXBEST,HLAM,C,A,COST,MOS)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*8 C(1100,0:1),A(1100,0:1)
INTEGER*2 IX(1),IXBEST(1),MOS(1)
COMMON CINF,BUDGET,TEST,NUMCEL
EPS=HLAM/1.DO5
ENDL=0.DO
ENDR=.01DO*HLAM
ZUBEST=CINF
AMBDA=(ENDL+ENDR)/2.DO
CALL MINFCN(AMBDA,ZL,IX,ZU,COST,C,A,MOS)
IF (COST.LE. BUDGET) THEN
  ENDR=AMBDA
  IF (ZU.LE. ZUBEST) THEN
    ZUBEST=ZU
    FLAM=AMBDA
  END IF
  GOTO 100
END IF
ENDL=AMBDA
ENDR=10.01DO*ENDR
AMBDA=(ENDL+ENDR)/2.DO
CALL MINFCN(AMBDA,ZL,IX,ZU,COST,C,A,MOS)

```



```

IF(COST .LE. BUDGET) THEN
  ENDR=AMBDA
  IF (ZU .LE. ZUBEST) THEN
    ZUBEST=ZU
    FLAM=AMBDA
  END IF
  GOTO 100
END IF
ENDL=AMBDA
ENDR=10.01DO*ENDR
100 AMBDA=(ENDL+ENDR)/2.DO
CALL MINFCN(AMBDA,ZL,IX,ZU,COST,C,A,MOS)
IF(COST .LE. BUDGET) THEN
  ENDR=AMBDA
  IF (ZU .LE. ZUBEST) THEN
    ZUBEST=ZU
    FLAM=AMBDA
  END IF
ELSE
  ENDL=AMBDA
END IF
IF ((ENDR-ENDL) .GT. EPS) GOTO 100
C CALCULATE BEST FEASIBLE SOLUTION
CALL MINFEA(FLAM,ZU,IXBEST,COST,C,A,IX,MOS)
RETURN
END
SUBROUTINE MINFCN(AMBDA,ZL,IX,ZU,COST,C,A,MOS)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*8 C(1100,0:1),A(1100,0:1)
INTEGER*2 IX(1),MOS(1)
COMMON CINF,BUDGET,TEST,NUMCEL
COST=0.DO
CELTOT=0.DO
ZU=0.DO
DO 400 I=1,NUMCEL
  IF (MOS(I) .LT. 0) THEN
    CELMIN=C(I,IX(I))+AMBDA*A(I,IX(I))
    COBJ=C(I,IX(I))
    INDEX=IX(I)
    GOTO 300
  END IF
  CELMIN=CINF
  INDEX=0
100 DO 200 J=0,5
    IF (C(I,J) .LE. TEST) THEN
      ZJ=C(I,J)+AMBDA*A(I,J)
      CELOBJ=C(I,J)
      IF (ZJ .LE. CELMIN) THEN
        IX(I)=J
        CELMIN=ZJ
        COBJ=CELOBJ
        INDEX=J
      END IF
    END IF
200 CONTINUE
300 CELTOT=CELTOT+CELMIN
    ZU=ZU+COBJ
    COST=COST+A(I,INDEX)
400 CONTINUE
    ZL=CELTOT-AMBDA*BUDGET
  RETURN
END
SUBROUTINE MINFEA(FLAM,ZU,IXBEST,COST,C,A,IX,MOS)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*4 (I-N)
REAL*8 C(1100,0:1),A(1100,0:1)
INTEGER*2 IXBEST(1),IX(1),MOS(1)

```

```

COMMON CINF,BUDGET,TEST,NUMCEL
COST=0.DO
CELTOT=0.DO
ZU=0.DO
DO 400 I=1,NUMCEL
  IF (MOS(I).LT. 0) THEN
    COBJ=C(I,IX(I))
    INDEX=IX(I)
    IXBEST(I)=IX(I)
    GOTO 300
  END IF
  CELMIN=CINF
  INDEX=0
100  DO 200 J=0,5
    IF (C(I,J).LE. TEST) THEN
      ZJ=C(I,J)+FLAM*A(I,J)
      CELOBJ=C(I,J)
      IF (ZJ.LE. CELMIN) THEN
        IXBEST(I)=J
        CELMIN=ZJ
        COBJ=CELOBJ
        INDEX=J
      END IF
    END IF
200  CONTINUE
300  ZU=ZU+COBJ
    COST=COST+A(I,INDEX)
400  CONTINUE
  RETURN
END
SUBROUTINE HEURIS(C,A,IXBEST,ZH,COST,MOS)
  IMPLICIT REAL*8 (A-H,O-Z)
  IMPLICIT INTEGER*4 (I-N)
  REAL*8 C(1100,0:1),A(1100,0:1),UARRAY(1100)
  INTEGER*2 IXBEST(1),MOS(1)
  COMMON CINF,BUDGET,TEST,NUMCEL
  COST1=COST
100  RESID=BUDGET-COST1
  DO 200 I=1,NUMCEL
    IF (MOS(I).LT. 0) THEN
      UARRAY(I)=-CINF
      GOTO 200
    END IF
    IF (IXBEST(I).GE. 5 .OR. C(I,(IXBEST(I)+1))
*    .GT. TEST) THEN
      UARRAY(I)=-CINF
      GOTO 200
    END IF
    RNUM=C(I,IXBEST(I))-C(I,(IXBEST(I)+1))
    RDENOM=A(I,(IXBEST(I)+1))-A(I,IXBEST(I))
    IF (RDENOM.GT. RESID .OR. RNUM.LE. 0.1D-08) THEN
      UARRAY(I)=-CINF
      GOTO 200
    ELSE
      UARRAY(I)=RNUM/RDENOM
    END IF
    IF (UARRAY(I).LT. 0.DO) UARRAY(I)=-UARRAY(I)
200  CONTINUE
  UBEST=-CINF
  DO 300 I=1,NUMCEL
    IF (UARRAY(I).GT. UBEST) THEN
      UBEST=UARRAY(I)
      INDEX=I
    END IF
300  CONTINUE
  IF (UBEST.LT. (-TEST)) GOTO 400
  COST1=COST1-A(INDEX,IXBEST(INDEX))
  **A(INDEX,(IXBEST(INDEX)+1))

```

```

IXBEST(INDEX)=IXBEST(INDEX)+1
IF (COST1 .LE. BUDGET) GOTO 100
400 ZH = 0.0D0
COST=0.0D0
DO 500 I=1, NUMCEL
    ZH=ZH+C(I, IXBEST(I))
    COST=COST+A(I, IXBEST(I))
500 CONTINUE
RETURN
END

```

APPENDIX B  
EXAMPLES OF INPUT FILES

PARAMETER FILE

```
          70000000
979      72000
         4
1010    12295  13405
        16000
0.75
1.00
```

# REENLISTMENT RATE FILE

18333	1	0.10	0.20	0.21	0.34	.43	.5
18333	2	0.65	0.52	0.69	0.88	.9	
18333	3	0.85	0.90	0.90	0.90		
21111	1	0.24	0.30	0.36	0.63	0.67	.7
21111	2	0.50	0.51	0.70	0.80	0.80	
21111	3	0.85	0.90	0.90	0.90		
21112	1	0.24	0.30	0.36	0.60	0.67	.7
21112	2	0.50	0.51	0.70	0.80	0.80	
21112	3	0.85	0.90	0.90	0.90		
21311	1	0.24	0.30	0.63	0.70	0.80	.85
21311	2	0.50	0.51	0.70	0.80	0.80	
21311	3	0.85	0.90	0.90	0.90		
21421	1	0.20	0.21	0.24	0.39	0.53	.63
21421	2	0.50	0.51	0.70	0.80	0.80	
21421	3	0.85	0.90	0.90	0.90		
21441	1	0.20	0.27	0.36	0.48	0.56	0.63
21441	2	0.50	0.51	0.70	0.80	0.80	
21441	3	0.85	0.90	0.90	0.90		
21451	1	0.20	0.21	0.37	0.38	0.65	.66
21451	2	0.50	0.51	0.70	0.80	0.80	
21451	3	0.85	0.90	0.90	0.90		
21461	1	0.24	0.30	0.36	0.62	0.65	.66
21461	2	0.50	0.51	0.70	0.80	0.80	
21461	3	0.85	0.90	0.90	0.90		
21471	1	0.24	0.30	0.36	.630	.67	.7
21471	2	0.50	0.51	0.70	0.80	0.80	
21471	3	0.85	0.90	0.90	0.90		
21611	1	0.20	0.30	0.41	0.42	0.43	0.63
21611	2	0.50	0.51	0.70	0.80	0.80	
21611	3	0.85	0.90	0.90	0.90		
21711	1	0.20	0.30	0.31	0.32	.43	0.57
21711	2	0.50	0.51	0.70	0.80	0.80	
21711	3	0.85	0.90	0.90	0.90		
21721	1	0.20	0.27	0.36	0.48	0.56	0.63
21721	2	0.50	0.51	0.70	0.80	0.80	
21721	3	0.85	0.90	0.90	0.90		
23111	1	0.23	.25	0.26	0.41	.53	.55
23111	2	0.53	0.70	0.71	0.72	.8	
23111	3	0.85	0.90	0.90	0.90		
23361	1	.21	.22	.5	0.94	1.00	1.
23361	2	0.86	.900	.910	.920	.95	
23361	3	0.85	0.90	0.90	0.90		
25121	1	0.21	0.39	0.40	0.70	.71	.75
25121	2	0.48	.75	.76	.88	.9	
25131	1	0.21	0.39	0.40	0.70	.71	.75
25131	2	.48	.75	.76	.88	.9	
25191	1	0.21	0.39	0.40	0.70	.71	.75
25191	2	0.77	.78	.8	.88	.9	
25191	3	0.85	0.90	0.90	0.90		
25311	1	0.21	0.39	0.40	0.70	.71	.75

# REENLISTMENT PLAN FILE

2629	1	1	22
2629	2	5	29
2629	3	9	12
2631	1	31	0
2631	2	2	8
2631	3	4	15
2632	1	4	10
2632	2	2	20
2632	3	3	13
2651	1	42	94
2651	2	4	0
2651	3	9	6
2671	1	10	0
2671	2	3	2
2671	3	1	6
2672	1	5	0
2672	2	1	0
2672	3	1	0
2673	1	8	0
2673	2	1	0
2673	3	1	4
2674	1	9	7
2674	2	1	0
2674	3	1	11
2675	1	10	16
2675	2	1	6
2675	3	2	5
2811	1	41	116
2811	2	30	0
2811	3	2	4
2813	1	20	27
2813	2	4	5
2813	3	1	0
2814	1	34	0
2814	2	6	6
2814	3	3	2
2818	1	49	0
2818	2	4	0
2818	3	3	0
2819	1	1	13
2819	2	1	13
2819	3	3	1
2822	1	3	10
2822	2	4	21
2822	3	6	9
2823	1	1	2
2823	2	1	9
2823	3	1	3
2825	1	6	6

# CELL SIZE FILE

0121	1	1098
0121	2	292
0121	3	4
0131	1	1151
0131	2	301
0131	3	11
0151	1	3567
0151	2	1036
0151	3	102
0161	1	194
0161	2	104
0161	3	38
0193	1	14
0193	2	422
0193	3	640
0211	1	18
0211	2	57
0211	3	24
0231	1	256
0231	2	221
0231	3	59
0241	1	24
0241	2	66
0241	3	30
0251	1	24
0251	2	76
0251	3	21
0311	1	13370
0311	2	1010
0311	3	34
0313	1	252
0313	2	62
0313	3	2
0331	1	2344
0331	2	252
0331	3	7
0341	1	2600
0341	2	238
0341	3	6
0351	1	2420
0351	2	245
0351	3	8
0352	1	1034
0352	2	252
0352	3	5
0369	2	1010
0369	3	831
0411	1	419
0411	2	176

# TRAINING COST FILE

0844	2	5211
0844	3	5211
0847	1	14027
0847	2	14027
0847	3	14027
0848	2	15600
0848	3	15600
0861	1	16000
0861	2	16000
0861	3	16000
1141	1	3256
1141	2	3256
1141	3	3256
1142	1	9865
1142	2	9865
1142	3	9865
1161	1	3484
1161	2	3484
1161	3	3484
1169	3	12775
1171	1	4386
1171	2	4386
1171	3	4386
1181	1	3500
1181	2	3500
1181	3	3500
1182	1	3500
1182	2	3500
1182	3	3500
1316	1	7041
1316	2	7041
1316	3	7041
1341	1	4536
1341	2	4536
1341	3	4536
1345	1	8853
1345	2	8853
1345	3	8853
1349	3	25216
1371	1	2826
1371	2	2826
1371	3	2826
1391	1	6705
1391	2	6705
1391	3	6705
1411	1	4386
1411	2	4386
1411	3	4386
1431	1	5311



# CELL WEIGHT FILE

2841	1	0	1.00
2841	2	0	1.00
2841	3	0	1.00
2861	1	0	1.00
2861	2	0	1.00
2861	3	0	1.00
2871	1	0	1.00
2871	2	0	1.00
2871	3	0	1.00
2874	1	0	1.00
2874	2	0	1.00
2874	3	0	1.00
2875	1	0	1.00
2875	2	0	1.00
2875	3	0	1.00
2881	1	0	1.00
2881	2	0	1.00
2881	3	0	1.00
2882	1	0	1.00
2882	2	0	1.00
2882	3	0	1.00
2884	1	0	1.00
2884	2	0	1.00
2884	3	0	1.00
2885	1	0	1.00
2885	2	0	1.00
2885	3	0	1.00
2886	1	0	1.00
2886	2	0	1.00
2886	3	0	1.00
2887	1	0	1.00
2887	2	0	1.00
2887	3	0	1.00
2889	1	0	1.00
2889	2	0	1.00
2889	3	0	1.00
3043	1	0	1.00
3043	2	0	1.00
3043	3	0	1.00
3044	1	0	1.00
3044	2	0	1.00
3044	3	0	1.00
3051	1	0	1.00
3051	2	0	1.00
3051	3	0	1.00
3052	1	0	1.00
3052	2	0	1.00
3052	3	0	1.00
3061	1	0	1.00

APPENDIX C  
SAMPLE FROM OUTPUT FILE

\*\*\*\*\*  
 OBJECTIVE FUNCTION VALUE IS: 1768.541  
 THIS SOLUTION IS WITHIN 0.00006802 % OF THE OPTIMAL  
 \*\*\*\*\*

MOS	ZONE	MULTIPLIER
121	1	1
121	2	0
121	3	1
131	1	1
131	2	0
131	3	1
151	1	2
151	2	0
151	3	1
161	1	5
161	2	0
161	3	0
193	1	5
193	2	4
193	3	0
211	1	5
211	2	4
211	3	1
231	1	5
231	2	0
231	3	1
241	1	5
241	2	0
241	3	0
251	1	5
251	2	4
251	3	1
311	1	1
311	2	4
311	3	1
313	1	5
313	2	0
313	3	1
331	1	5
331	2	4
331	3	1
341	1	1
341	2	0
341	3	1
351	1	0
351	2	0
351	3	0
352	1	0
352	2	0
352	3	1
369	2	4
369	3	0

APPENDIX D  
USER INSTRUCTIONS

A. DATA FILES

Input to the program is provided through six data files. File names, specifications and entries are detailed in the accompanying tables. In general, data values should be right justified within the fields. Rates, percentages, and the exogenous weighting factors may be placed anywhere within the specified field provided they are entered as two place decimals (e.g., .25, .03 or 1.15). With the exception of the Parameter file, all files contain the same entries in the first 8 columns. The program reads data values sequentially from the files. Therefore, it is vital that the MOSs, Zones and accompanying entries from one file correspond line for line with all other files. For example, if line 15 from one file contains entries pertaining to MOS 0193, Zone C, then line 15 from all other files should refer to MOS 0193, Zone C.

Those cells for which a non-zero SRB multiplier is expressly prohibited, such as the cells in the Band and Marine Corps Exchange MOSs, should be eliminated from the data files. Similarly, there are MOSs held only by very senior Marines (E-8's and E-9's) not eligible for bonuses due to years-of-service requirements exceeding 14 years. These, too, should be purged from the data files. As written, SRBMULT.FOR can read a maximum of 1100 cells. If at some point it becomes necessary to increase this, it will require changing the array and matrix dimensions within the program from 1100 to the desired number.

TABLE 1a  
DATA FILE SPECIFICATIONS

Reenlistment Plan File (RPLAN.DAT)

Line	Column	Data Description
all	1-5	MOS
	6-7	blank
	8	Zone (1 for A, 2 for B, 3 for C)
	9-18	Number of Marines eligible for reenlistment in cell
	19-28	Reenlistment target for cell

TABLE 1b  
DATA FILE SPECIFICATIONS

Cell Size File (ACTNUM.DAT)

Line	Column	Data Description
all	1-5	MOS
	6-7	blank
	8	Zone (1 for A, 2 for B, 3 for C)
	9-14	Total number of Marines in cell

TABLE 1c  
DATA FILE SPECIFICATIONS

Training Cost File (TCOST.DAT)

Line	Column	Data Description
all	1-5	MOS
	6-7	blank
	8	Zone (1 for A, 2 for B, 3 for C)
	9-19	Training cost (dollars)

TABLE 1d  
DATA FILE SPECIFICATIONS

Response Rate File (RRATE.DAT)

Line	Column	Data Description
all	1-5	MOS
	6-7	blank
	8	Zone (1 for A, 2 for B, 3 for C)
	9-14	Response to SRB multiplier of 0
	15-20	Response to SRB multiplier of 1
	21-26	Response to SRB multiplier of 2
	27-32	Response to SRB multiplier of 3
	33-38	Response to SRB multiplier of 4 (if applicable)
	39-44	Response to SRB multiplier of 5 (if applicable)

TABLE 1e  
DATA FILE SPECIFICATIONS  
Cell Weight File (WEIGHT.DAT)

Line	Column	Data Description
all	1-5	MOS
	6-7	blank
	8	Zone (1 for A, 2 for B, 3 for C)
	9-10	blank
	11	Preset SRB multiplier (0 if not preset)
	12-13	blank
	14-18	Exogenous weighting factor (typically, use 1.00. If other values are used, include decimal, e.g. .50)

TABLE 1f  
DATA FILE SPECIFICATIONS  
Parameter File (PARAM.DAT)

Line	Column	Data Description
1	1-12	Budget available for current year bonuses (dollars)
2	1-4	Total number of cells
3	1-12	Maximum training cost (dollars)
4	1-7	Zone A avg reenlistment duration
	8-14	Zone B avg reenlistment duration
	15-21	Zone C avg reenlistment duration (years)
5	1-7	Zone A avg monthly base pay
	8-14	Zone B avg monthly base pay
	15-21	Zone C avg monthly base pay (dollars)
6	1-12	Maximum permissible bonus (dollars)
7	1-4	Percentage of bonus awarded as lump sum (in decimal form, e.g., .75)
8	1-4	Over/under factor

## B. RUNNING THE PROGRAM

To use the program, the user should possess, at a minimum, the IBM PC configuration detailed in Chapter IV and a FORTRAN 77 compiler. The program was developed and tested using the Ryan-McFarland RM/FORTRAN compiler. However, the code used in the program is sufficiently generic that other FORTRAN 77 compilers should be acceptable.

A diskette ("program diskette") containing the following files will be provided to the Enlisted Plans Section, Manpower Department, HQMC: HOWTO.DOC, RUNCHK.BAT, FILCHK.FOR, FILCHK.EXE, RUNSRB.BAT, SRBMULT.FOR, and SRBMULT.EXE. The file HOWTO.DOC contains a copy of operating instructions. Files with filetype .BAT are executive routines, those with filetype .FOR are FORTRAN source code and files with filetype .EXE are executable programs. Use of these programs is detailed in the following sections. Prior to running any of these programs certain parameters within the DOS configuration file CONFIG.SYS must be changed. If this file does not already exist, it will be necessary to create it. The RM/FORTRAN that was used to compile SRBMULT.EXE and FILCHK.EXE requires that the number of files that can be opened concurrently be increased to at least ten. Therefore, the following line must be added to CONFIG.SYS:

FILES = 10

Similarly, the number of disk buffers allocated by DOS at system startup must be increased to ten. Accordingly, add:

BUFFERS = 10

to CONFIG.SYS. Once these changes have been made to CONFIG.SYS, reboot DOS.

### 1. Checking Data Files for Conformity

The FORTRAN program FILCHK.EXE is provided to check the files for MOS and Zone conformity prior to using SRBMULT.FOR. To use FILCHK.EXE on a system with two disk drives, place the program diskette in B drive and the diskette containing the data files ("data diskette") in A drive. Enter the following DOS commands:

```
Path = B:\
```

```
A:
```

The first command will cause DOS to check the B drive for executable files. The second sets the A drive as the default drive. At the system prompt, enter RUNCHK to execute the program RUNCHK.BAT. This executive routine will set necessary file definitions and then execute FILCHK.EXE. On a system with a hard disk, place all files in the same directory on the C drive (hard disk) and then execute RUNCHK.BAT as before.

The FILCHK program uses the sequence of MOSs and Zones in the file WEIGHT.DAT as a reference set and checks the other four data files for conformance to it. If a non-conforming entry is discovered, the applicable file definition number and line number are written to the screen. The remaining files are then checked. Once detected errors are corrected, RUNCHK.BAT should be executed again until all files conform with the reference as evidenced by no error returns.

### 2. Running SRBMULT

To run SRBMULT.EXE on a system with two disk drives, place the program diskette in the B drive and the data diskette in the A drive. Enter the following DOS commands:

```
PATH = B:\
```

```
A:
```

```
RUNSRB
```



Several minutes later, an output file MULPLN.DAT will be created on the diskette in the A drive. It will contain an MOS and Zone listing, the recommended SRB multipliers, the objective function value and the percent deviation from the optimal. The objective function value is useful in judging the effect of introducing exogenous weighting factors or presetting multipliers. Instructions on how to do both of these is contained in the following section.

Running the program is somewhat simpler on a system with a hard disk. All files should be placed in a single directory on the C drive (hard disk) and that drive should be made the default drive. SRBMULT.EXE is initiated by typing RUNSRB to execute RUNSRB.BAT.

C. EXOGENOUS WEIGHTING FACTORS, PRESET MULTIPLIERS AND THE OVER/UNDER FACTOR

Inclusion of an exogenous weighting factor (columns 14-18 in WEIGHT.DAT) permits the user to exercise additional control in weighting the effects of reenlistment shortfalls in particular cells based on his judgement. Weighting factors  $W_i$  greater than the default value of 1.0 result in comparatively greater penalties for failing to meet reenlistment targets. Conversely, weighting factors less than 1.0 result in lesser penalties. The effect of the factor is linear. Thus a 2.0 results in a penalty twice as large as a 1.0 and a factor of .5 reduces the penalty by one-half. As a weighting scheme based on training costs and manning levels is already utilized, this capability should be used with caution. In general, a value of 1.0 should be used for all cells.

The user can also preset SRB multipliers. This might be necessary if factors other than those considered by the model dictate that particular cells be assigned specific multipliers. This is accomplished by entering a minus sign (-) in column 1 of the pertinent cell's data entry in the

file WEIGHT.DAT. The desired multiplier is then entered in column 11 of the same line. In the output file, the minus sign will appear next to the appropriate MOS/Zone thereby "flagging" those cells with predetermined multipliers.

The last entry in PARAM.DAT is the over/under factor Q, the weighting factor employed to permit the user to establish the relative importance of overshooting or undershooting reenlistment targets. For example, a over/under factor of .70 implies that the penalty incurred in the model for overshooting the reenlistment target by some number is only 70% of the penalty that would be incurred for undershooting the target by the same number. Since an overage in a particular cell is generally considered more acceptable than a shortage, the factor will typically be less than 1.

# LIST OF REFERENCES

1. DoD Instruction 1304.22, *Administration of Enlisted Personnel Bonus and Special Duty Assignment Pay Programs*, August 7, 1985.
2. Muckstadt, J. and Koenig, S. A., "An Application of Lagrangian Relaxation to Scheduling in Power Generation Systems," *Operations Research*, v. 25, pp. 387-403, 1977.
3. Geoffrion, A. M. and McBride, R., "Lagrangian Relaxation Applied to Capacitated Facility Location Problems," *AIIE Transactions*, v. 10, pp. 40-47, 1978.
4. Etcheberry, J., "The Set-Covering Problem: A New Implicit Enumeration Algorithm," *Operations Research*, v. 25, pp. 760-772, 1977.
5. Kostik, P., Analyst, Center for Naval Analysis, Alexandria, Virginia, Interview, December 23, 1985.
6. Greenberg, H. and Hegerich, R. L., "A Branch Search Algorithm for the Knapsack Problem," *Management Science*, v. 16, pp. 327-332, January 1970.
7. Dreyfus, S. E. and Law, A. M., *The Art and Theory of Dynamic Programming*, pp. 107-118, Academic Press, 1977.
8. Ross, G. T. and Soland, R. M., "Modeling Facility Location Problems as Generalized Assignment Problems," *Management Science*, v. 24, pp. 345-357, November 1977.
9. Ross, G. T. and Soland, R. M., "A Branch and Bound Algorithm for the Generalized Assignment Problem," *Mathematical Programming*, v. 8, pp. 118-132, 1975.
10. Fisher, M. L., "An Applications Oriented Guide to Lagrangian Relaxation," *Interfaces*, v. 15, pp. 10-21, March-April 1985.
11. Geoffrion, A. M., "Lagrangian Relaxation for Integer Programming," *Mathematical Programming Study* 2, pp. 82-114, 1974.

# INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2	
2. Library, Code 0142 Naval Postgraduate School Monterey CA 93943	2	
3. Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey CA 93943	1	
4. Professor R. Kevin Wood Code 55Wd Department of Operations Research Naval Postgraduate School Monterey CA 93943	12	
5. Professor Richard E. Rosenthal Code 55R1 Department of Operations Research Naval Postgraduate School Monterey CA 93943	5	
6. Professor Paul Milch Code 55Mh Department of Operations Research Naval Postgraduate School Monterey CA 93943	1	
7. Commandant of the Marine Corps Code MPP-24 Headquarters United States Marine Corps Washington D.C. 20380-0001	1	
8. Commandant of the Marine Corps Code MPI-40 Headquarters United States Marine Corps Washington D.C. 20380-0001	1	
9. CAPT Dean D. DeWolfe 8505 A Barrington Court Springfield VA 22152	5	

END

DTTC

7-86